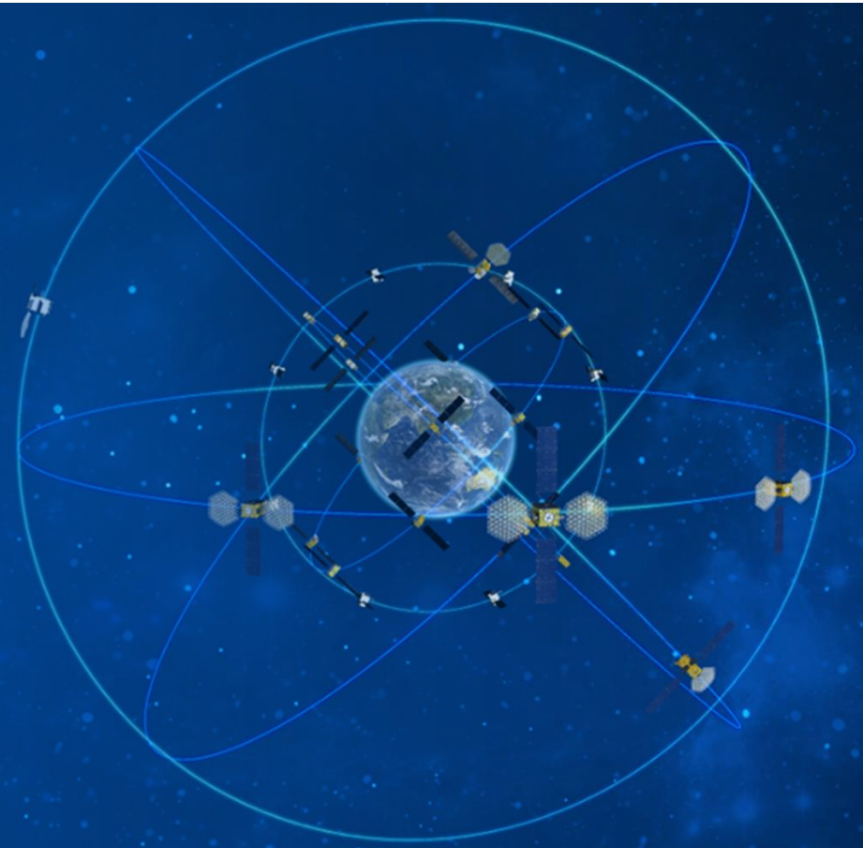
Molecular property prediction

分子性质预测



2252933 李海禄

目录

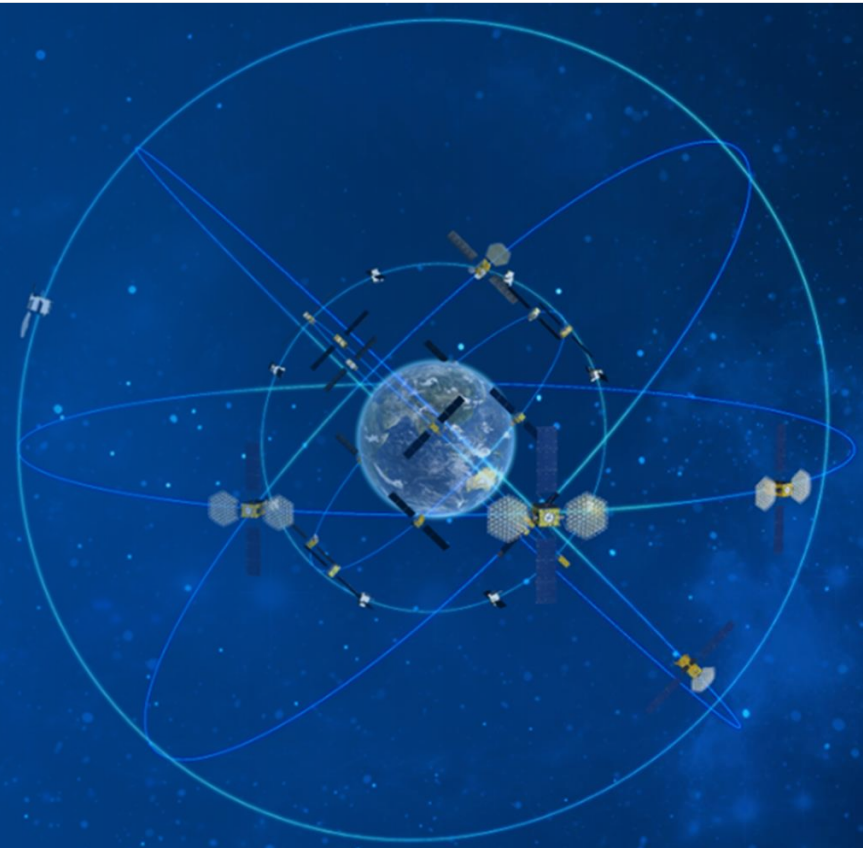
CONTENTS

Molecular Classification

PART ONE

研究背景

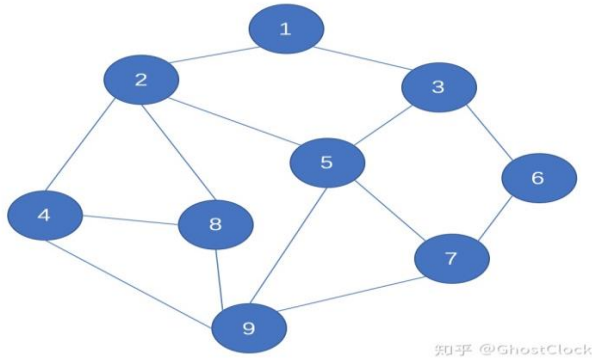
图结构是描述和分析实体与关系的一种通用语言，被普遍应用在许多领域中，如社交网络、交通网络、引文网络和分子结构等。对图结构数据来说，一个基本问题是计算它们之间的相似性，以便进行图分类

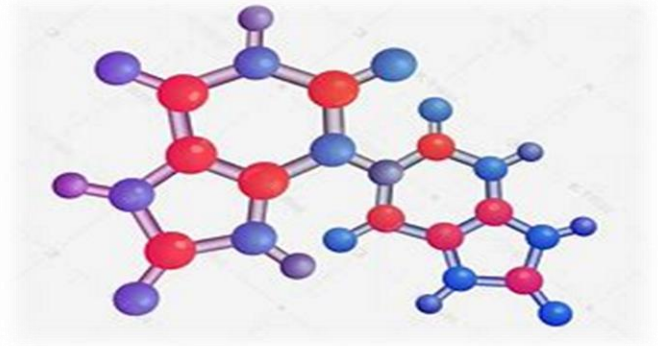


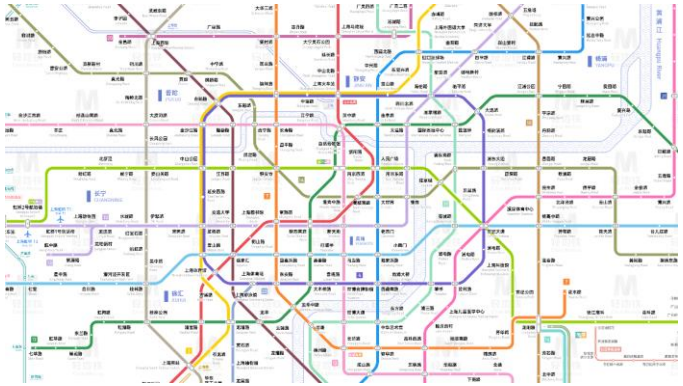
引言

项目背景

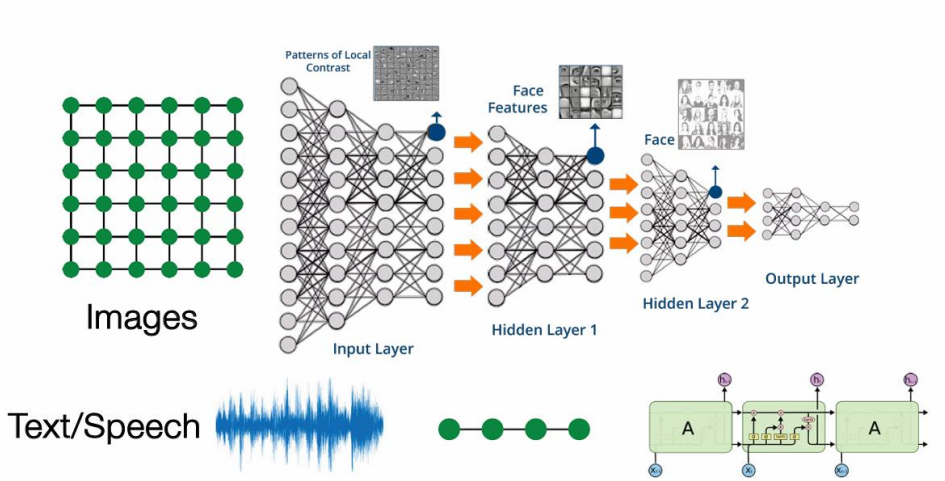
- 图结构是描述和分析实体与关系的一种通用语言，被普遍应用在许多领域中，如社交网络、交通网络、引文网络和分子结构等。
- 对图结构数据来说，一个基本问题是计算它们之间的相似性，以便进行图分类



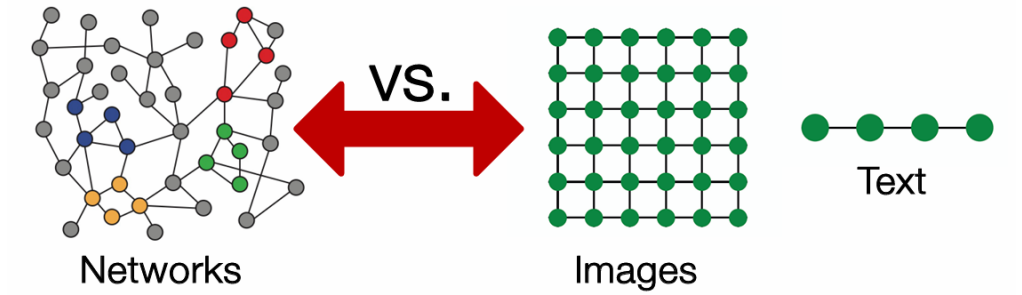




现代深度学习框架



数据结构对比



项目动机

- 现代深度学习的框架大多为处理较简单的序列或者网格数据而设计
- 但在图上远远没那么简单：任意大小和复杂的拓扑结构、没有像网格那样的空间局部性、没有固定顺序和参考节点、甚至是动态变化的

研究主题

选取分子性质预测为研究主题，分子性质预测在药物筛选和药物设计等中有着重要作用。其核心在于利用分子的内部信息来预测其物理和化学性质，来达到筛选作用，显著加速药物研发进程。

传统上，DFT方法被广泛应用于分子性质预测，但这种方法非常耗时且计算成本高昂。

近年来，深度学习方法为分子性质预测提供了新的思路。将分子视为图数据，分子中的原子被抽象为节点，化学键被抽象为边，构建出分子图，结合深度学习方法进行预测。

这样，分子性质预测问题就转化为了一个图回归问题。

数据库选择

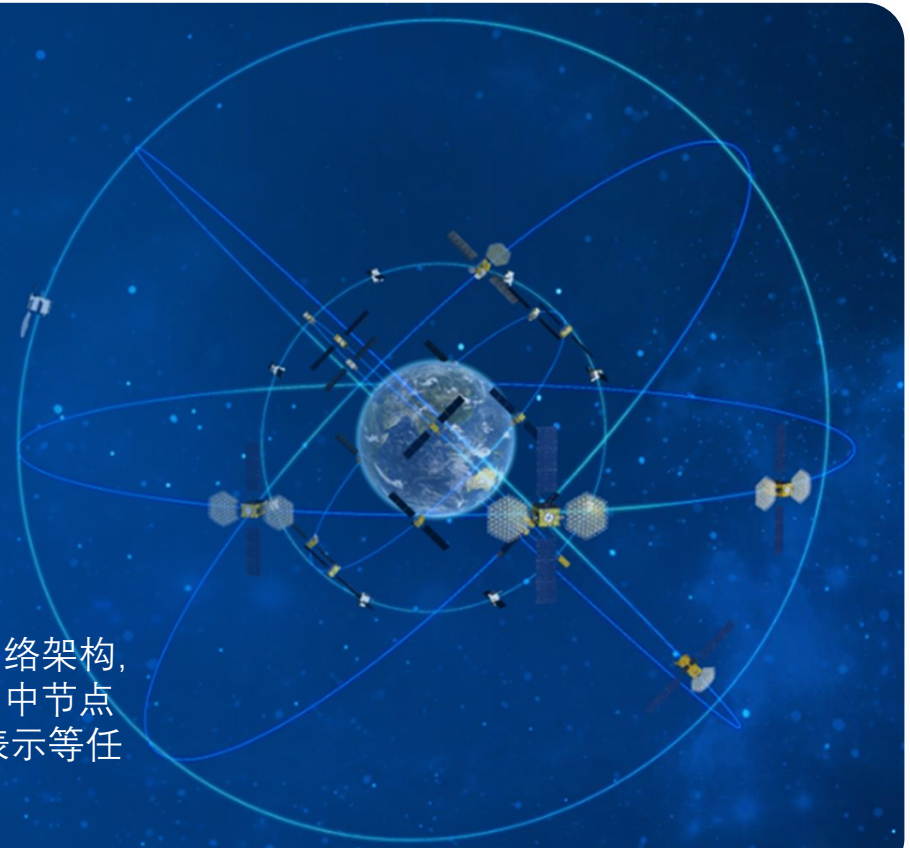
TUDatasets

这是TU Dortumnd University收集的大量关于分子特征的图数据，可以通过PyTorch Geometric直接加载。

例如其中一个数据集MUTAG是一个硝基芳香族化合物的集合，目的是预测它们对鼠伤寒沙门氏菌的诱变性。输入图用于表示化合物，其中顶点代表原子并由原子类型标记（由one-hot编码表示），而顶点之间的边表示相应原子之间的键。它包括188个化合物样品，有7个离散节点标签。

PART TWO

GCN模型



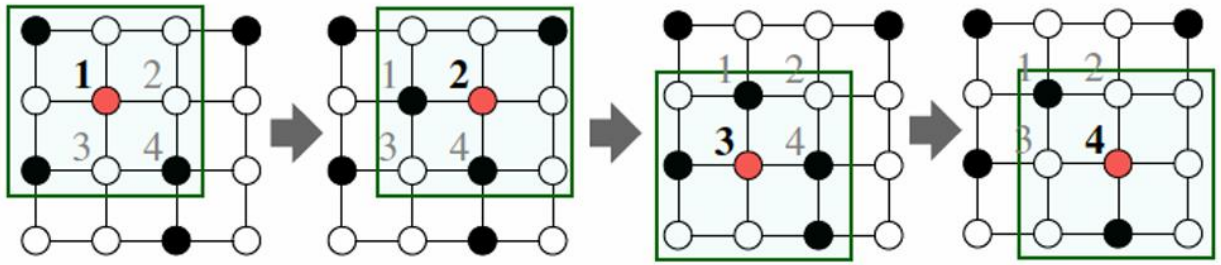
GCN（图卷积网络）是一种在图结构数据上进行卷积操作的神经网络架构，通过聚合邻居节点的信息来更新每个节点的特征。GCN能够捕获图中节点间的复杂关系和特征，用于节点分类、图分类、边预测及图嵌入表示等任务。

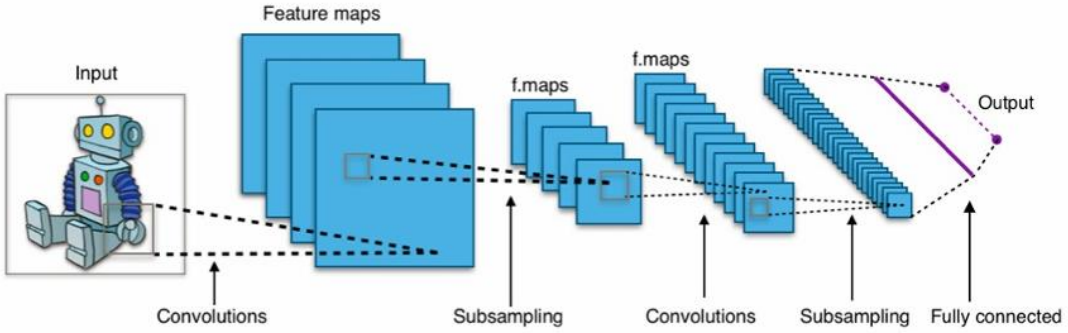
构建模型

之

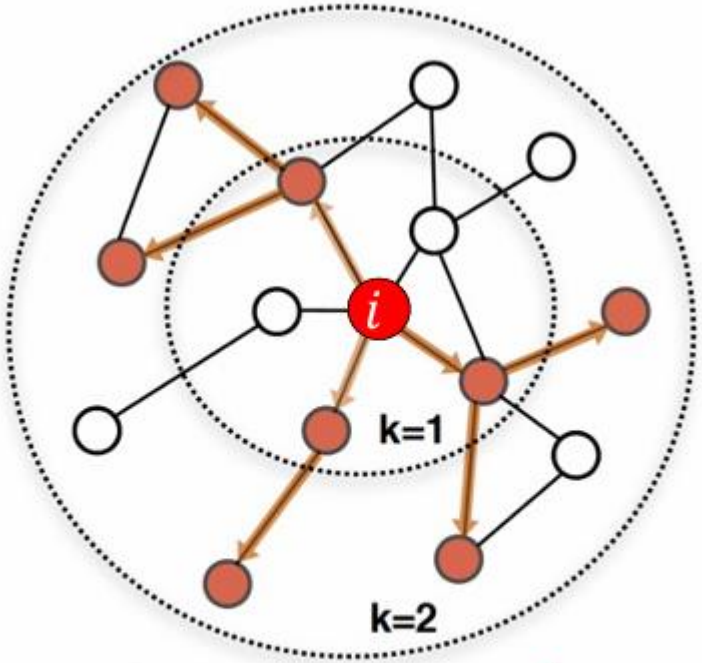
图卷积网络

CNN on an image:



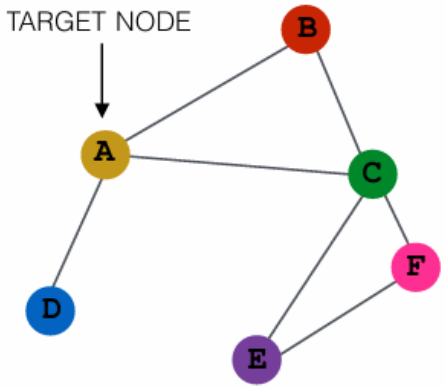


在图结构为以每个结点的领域作为计算图获取结构信息

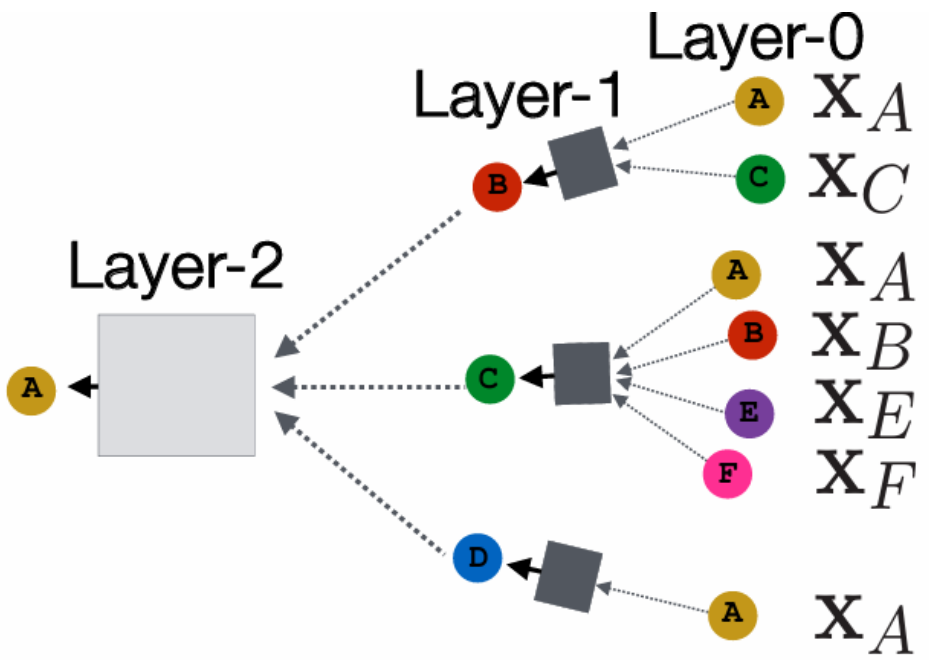


Determine node computation graph

图卷积层



INPUT GRAPH



$$h_v^{(k+1)} = \sigma(W_k \sum_{u \in N(v)} \frac{h_u^{(k)}}{N(v)} + B_k h_v^{(k)})$$

单层感知器

线性变换+激活函数

聚合函数

使用ReLU函数

图读出层

读出阶段是通过全局的池化模块来生成整个图级的特征。在卷积阶段过后得到最终图节点特征。

$$\mathrm{H}^k = \{h_1^k, h_2^k, \ldots, h_N^k\}$$

$$\mathrm{R}(\mathrm{H}^k) = \sigma(P(\mathrm{H}^k))$$

单层感知器

池化函数

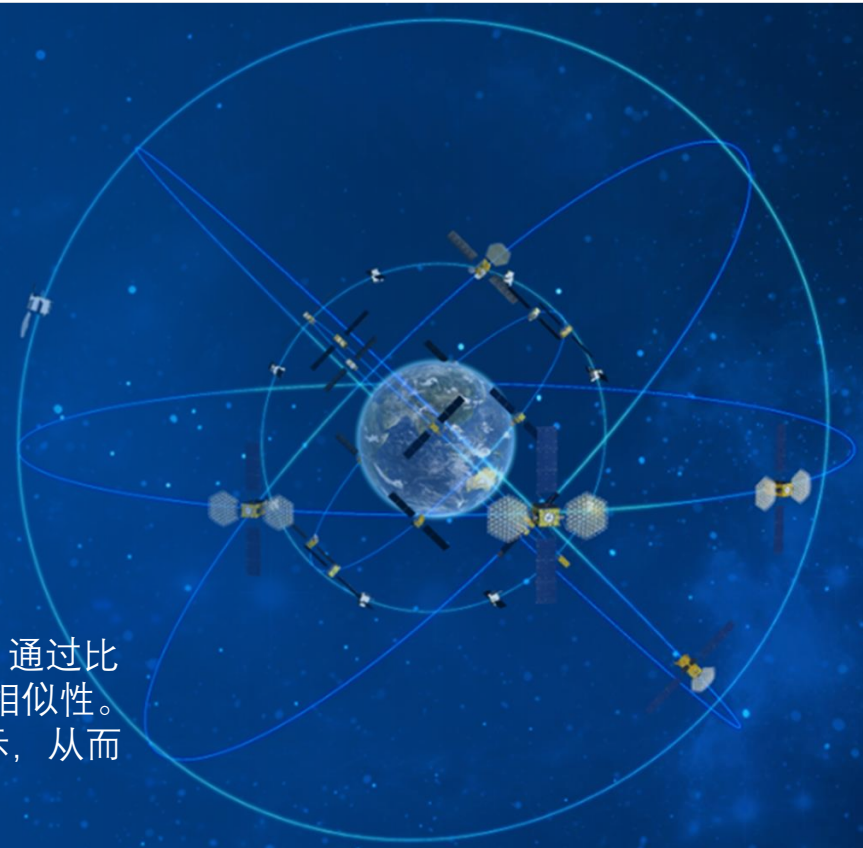
本项目读出函数使用log函数作为激活函数，P采用全局平均化池：

$$\sigma(X) = log\,(AX + b)$$

$$P(\mathrm{H}^k) = \sum_{i \in [1,N]} \frac{h_{\mathrm{i}}^{(k)}}{N}$$

PART THREE

WL核模型



WL核利用Weisfeiler-Lehman图同构测试算法迭代地丰富节点特征，通过比较两个图在WL算法迭代过程中的节点特征向量来计算它们之间的相似性。这种方法的核心在于利用图的邻域结构信息来构建节点的特征表示，从而捕捉图的结构特性。

构建模型

之

1阶-WL算法

1.节点多重集标签聚合

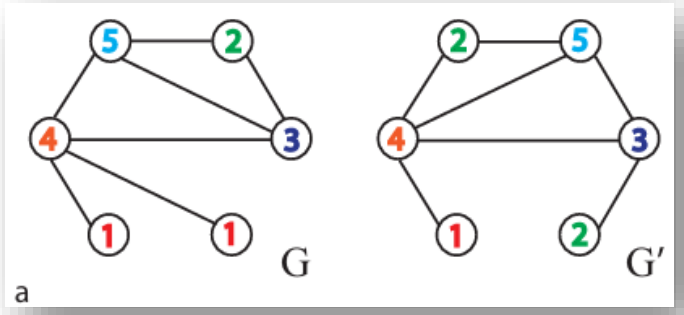
对每个节点将其领域节点的标签聚集为一个有序多重集序列，将该节点原本标签放在序列最前面得到标签

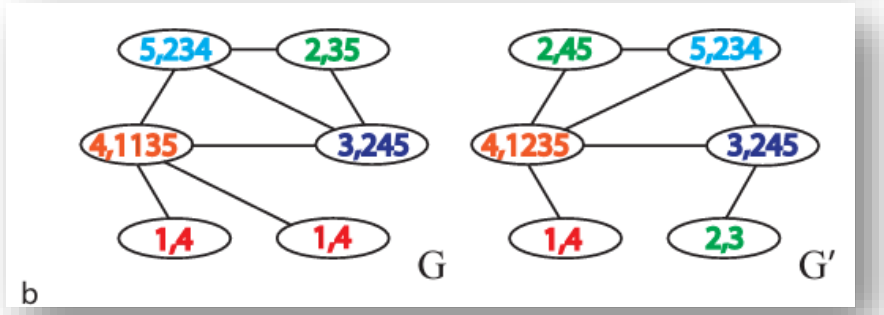
2.节点标签压缩简化

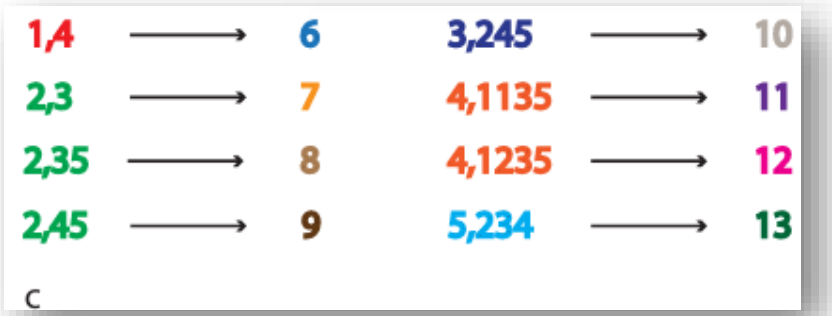
将第一步得到的标签用单射hash映射为一个新的整数标签。

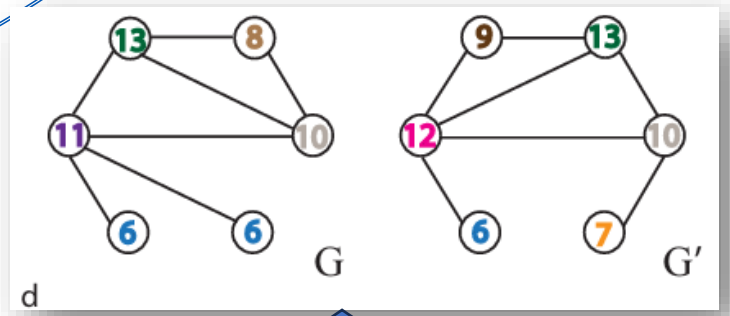
3.得到新的特征图

将所有节点的标签换为第二步得到的标签得到新的特征图

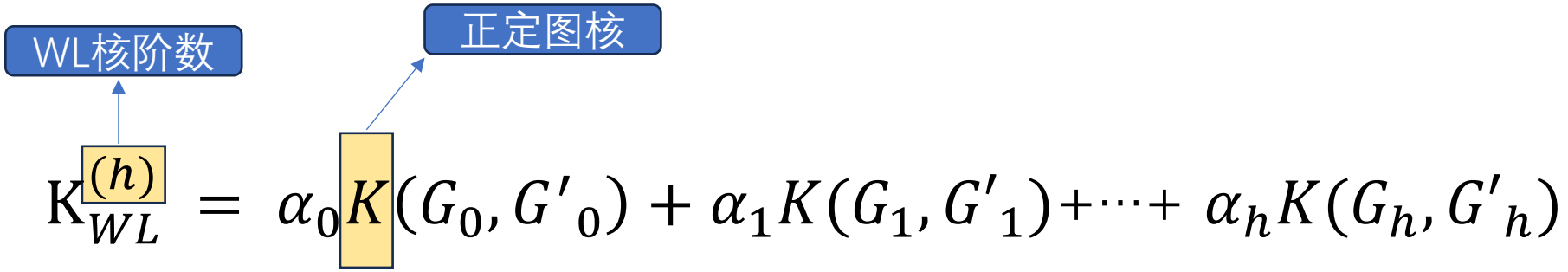








WL核函数

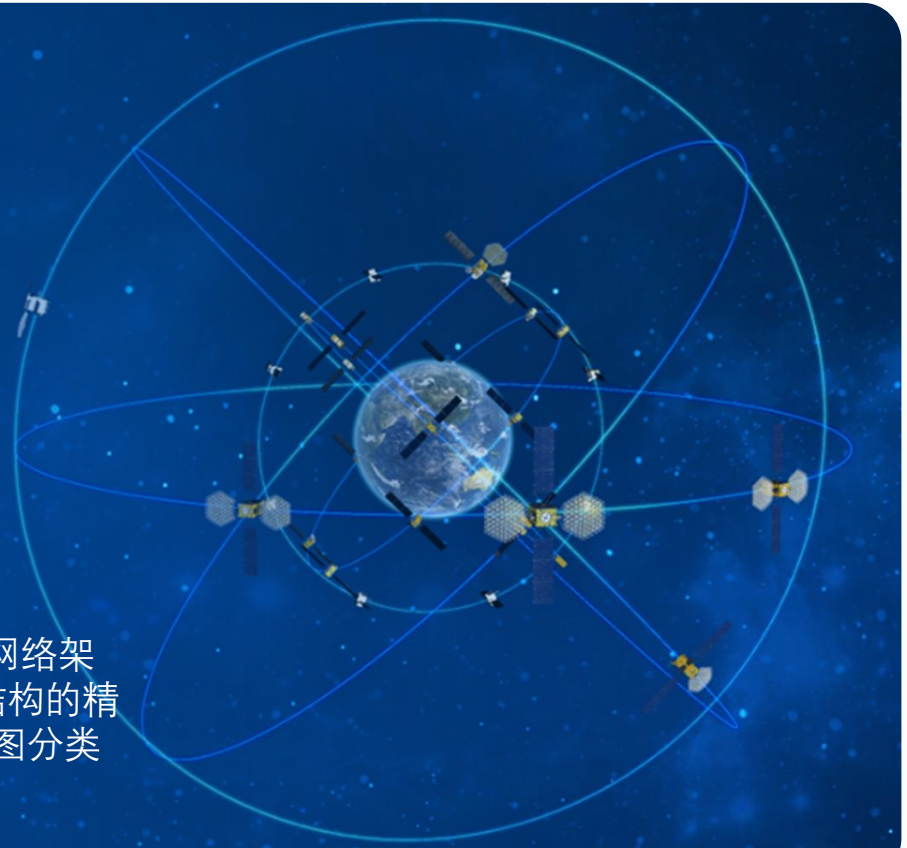


在使用WL核表达出图之间的相似性后，使用SVM来进行分类，直接使用WL核函数来替换SVM中的内积计算，这样SVM就可以利用WL核函数来度量图之间的相似性，并进行分类。

这里K是自己选定的常用的正定核函数，如子树核，最短路径核，随机游走核等，h是自己通过经验感知选定的合适的阶数。

PART FOUR

GIN模型



图同构网络（GIN）模型是一种比GCN拥有更强表达能力的图神经网络架构，它通过迭代聚合节点邻居特征并混合自身特征，实现了对图结构的精确表达。GIN模型具有单射特性，能够区分不同的图结构，适用于图分类等任务。

HOW POWERFUL ARE GNN?

GNN的设计是一个非常经验感觉的设计，如何客观形式化地探知其在图上的表达能力，在《HOW POWERFUL ARE GNN?》这篇文章中给出了说明。

GNN的表达能力的严格上限是WL同构检验

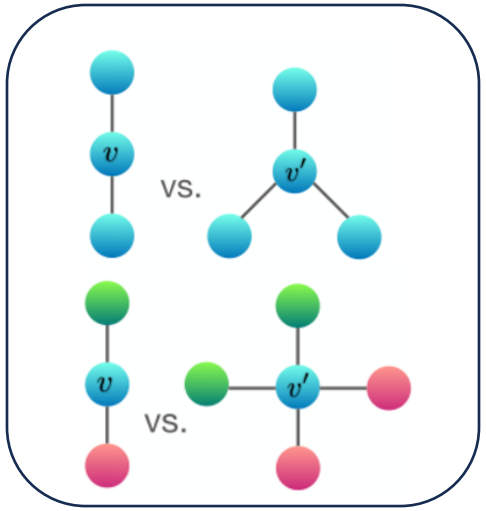
- 相同：都是一阶一阶迭代聚合得到结点丰富特征。
- 不同：WL test 是由哈希函数聚合，GNN 池化函数来聚合。

前者是单射函数，可以保证不同构的图映射到不同的embedding中，但池化函数并不是单射

可以保证单射聚合的GNN变体：GIN

1.采用单射聚合函数：sum

2.图级读出函数保证单射



Max和mean池化无法区分

核方法VS神经网络

•核方法的选择和参数调整对性能有较大影响，需要一定的经验和技巧。对于大规模数据集，核方法的计算效率较低。

•神经网络通过训练可以自动学习调整参数，神经网络在处理大规模数据集时具有较高的效率。

构建模型

之

图同构网络模型

GCN使用单层感知机，其本质上是线性模型，可能无法捕捉到数据中复杂的非线性关系，从而限制了GCN的表达能力。这里将ReLU激活函数替换为MLP。

Sum聚合函数

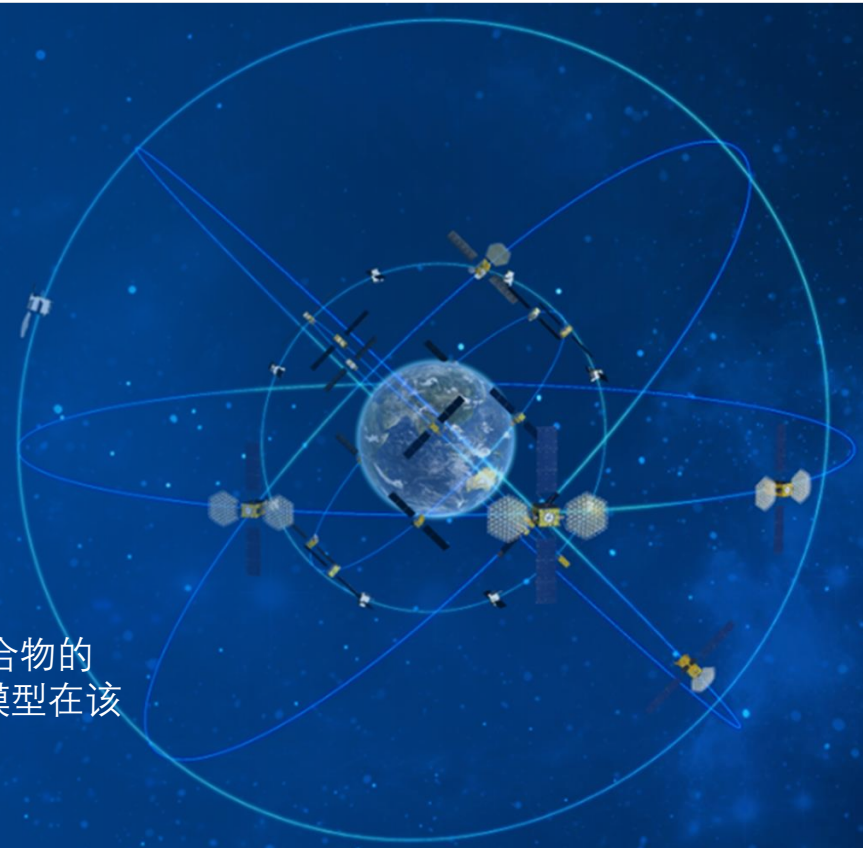
$$h_v^{(k+1)} = MLP^{(k+1)}\left(\sum_{u \in N(v)} h_u^{(k)} + (1+\epsilon^k) h_v^{(k)}\right)$$

Readout函数

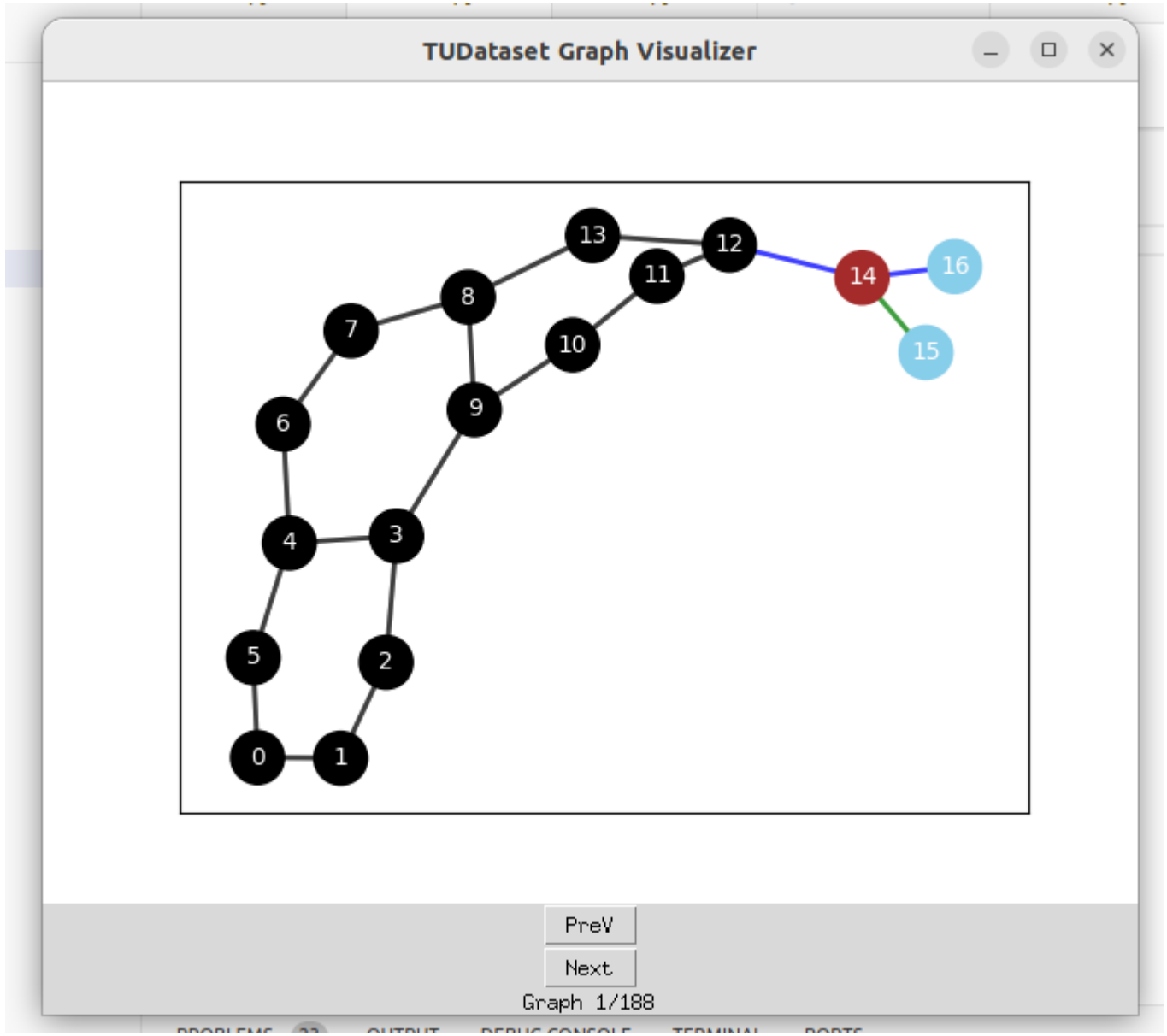
$$\mathrm{R}\left(\mathrm{H}^k\right) = Concat\left(\left(\sum_{i \in [1,N]} h_i^{(k)}\right) \middle| k = 0,1,\ldots,K\right)$$

PART FIVE

实验部分

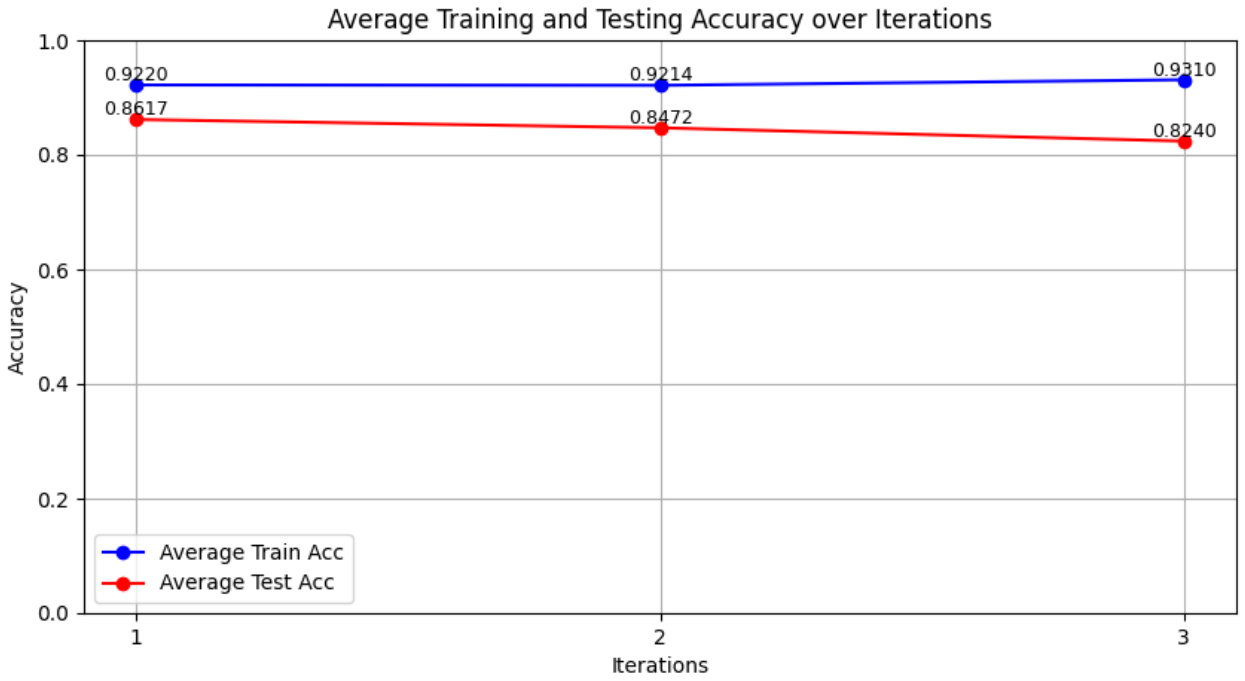


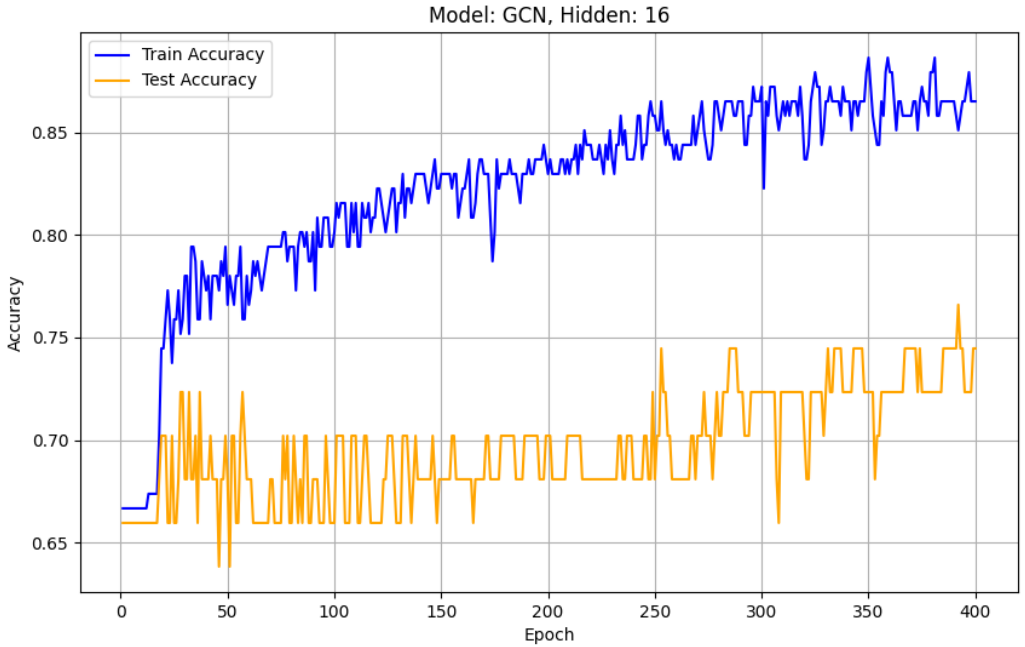
使用三种模型对数据集MUTAG分类，MUTAG是一个硝基芳香族化合物的集合，目的是预测它们对鼠伤寒沙门氏菌的诱变性。对比了三种模型在该分类任务中的实际效果。

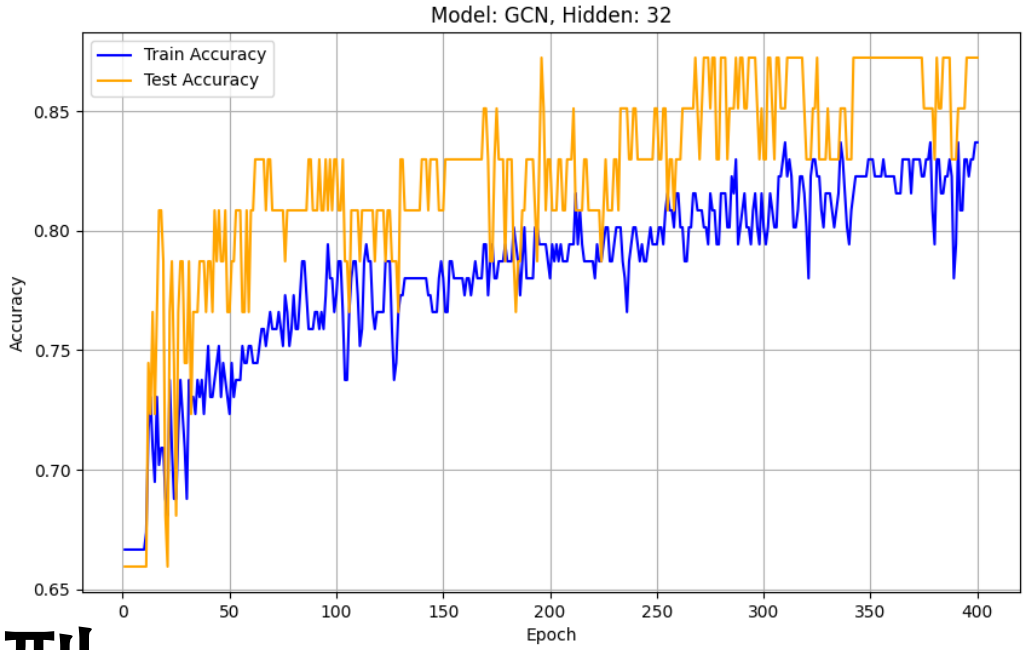

TUDataset Graph Visualizer
13
12
11
14
16
8
7
10
15
9
6
4
3
5
2
0
1
PreV
Next
Graph 1/188

```
iter=1
Model: WL_SVM, Train Acc: 0.8511, Test Acc: 0.8511
Model: WL_SVM, Train Acc: 0.9149, Test Acc: 0.8723
Model: WL_SVM, Train Acc: 0.9645, Test Acc: 0.8298
Model: WL_SVM, Train Acc: 0.8582, Test Acc: 0.8511
Model: WL_SVM, Train Acc: 0.9433, Test Acc: 0.8723
Model: WL_SVM, Train Acc: 0.9504, Test Acc: 0.8723
Model: WL_SVM, Train Acc: 0.9504, Test Acc: 0.8298
Model: WL_SVM, Train Acc: 0.9078, Test Acc: 0.8511
Model: WL_SVM, Train Acc: 0.9362, Test Acc: 0.9149
Model: WL_SVM, Train Acc: 0.9433, Test Acc: 0.8723
iter=2
Model: WL_SVM, Train Acc: 0.9362, Test Acc: 0.8723
Model: WL_SVM, Train Acc: 0.9504, Test Acc: 0.8936
Model: WL_SVM, Train Acc: 0.8440, Test Acc: 0.8511
Model: WL_SVM, Train Acc: 0.9149, Test Acc: 0.8085
Model: WL_SVM, Train Acc: 0.9574, Test Acc: 0.8936
Model: WL_SVM, Train Acc: 0.9220, Test Acc: 0.8298
Model: WL_SVM, Train Acc: 0.9504, Test Acc: 0.8085
Model: WL_SVM, Train Acc: 0.9220, Test Acc: 0.7872
Model: WL_SVM, Train Acc: 0.9574, Test Acc: 0.8511
Model: WL_SVM, Train Acc: 0.9362, Test Acc: 0.8511
Model: WL_SVM, Train Acc: 0.8440, Test Acc: 0.8723
iter=3
Model: WL_SVM, Train Acc: 0.8511, Test Acc: 0.7660
Model: WL_SVM, Train Acc: 0.9149, Test Acc: 0.8723
Model: WL_SVM, Train Acc: 0.9645, Test Acc: 0.8298
Model: WL_SVM, Train Acc: 0.9504, Test Acc: 0.7872
Model: WL_SVM, Train Acc: 0.9362, Test Acc: 0.9149
Model: WL_SVM, Train Acc: 0.9716, Test Acc: 0.7872
Model: WL_SVM, Train Acc: 0.9504, Test Acc: 0.8298
Model: WL_SVM, Train Acc: 0.9504, Test Acc: 0.8723
Model: WL_SVM, Train Acc: 0.8511, Test Acc: 0.8085
Model: WL_SVM, Train Acc: 0.9362, Test Acc: 0.7872
Model: WL_SVM, Train Acc: 0.9645, Test Acc: 0.8085
```

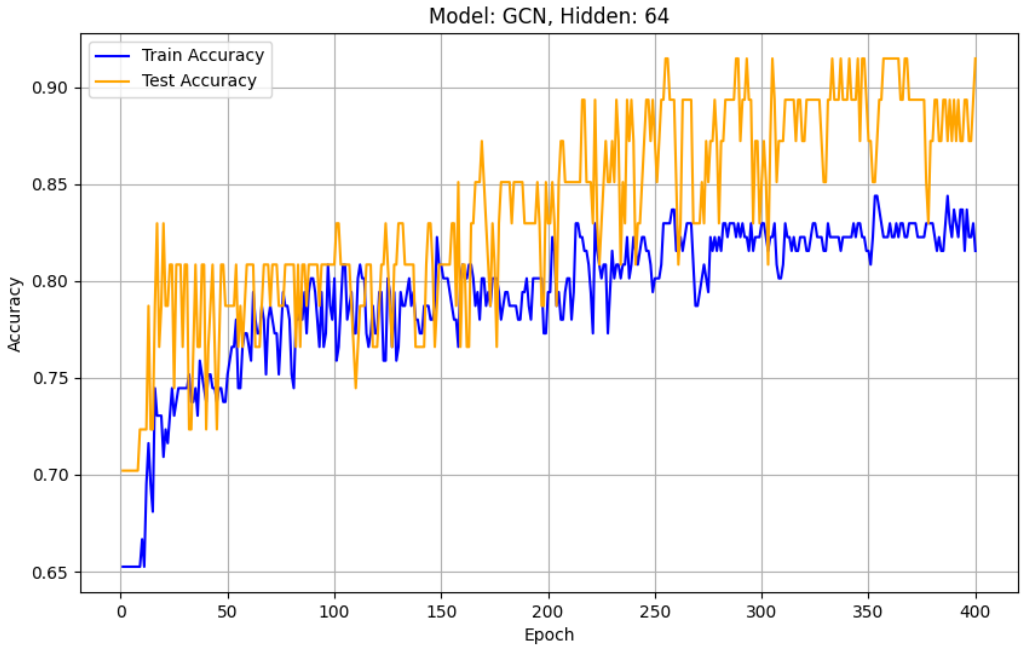
WL核_SVM模型

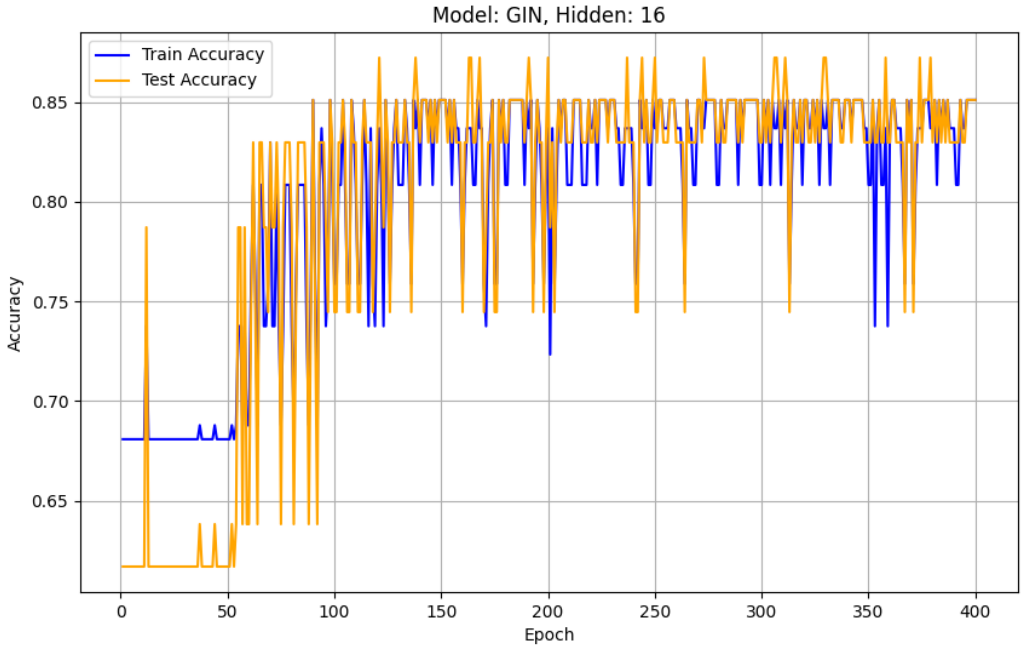


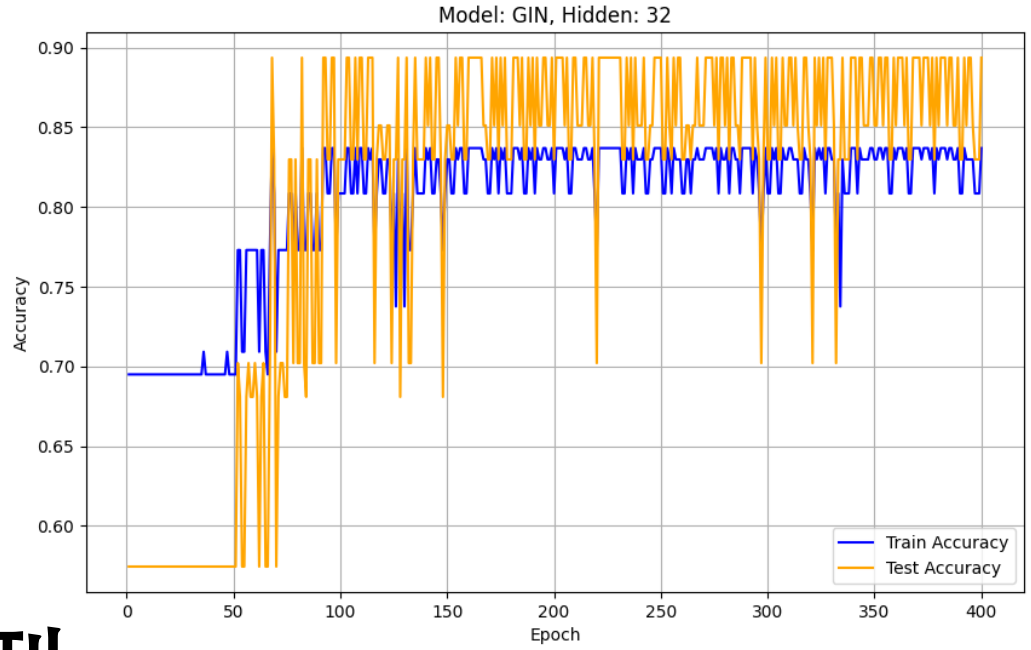

Model: GCN, Hidden: 16
Train Accuracy
Test Accuracy
Accuracy
Epoch


Model: GCN, Hidden: 32
Train Accuracy
Test Accuracy
Accuracy
Epoch

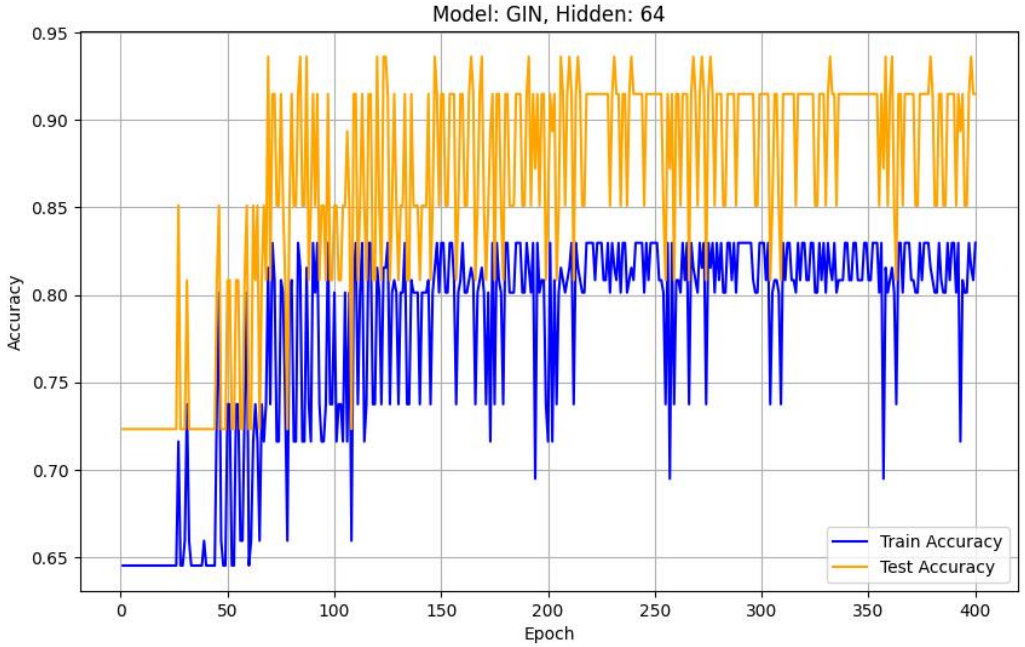
GCN模型

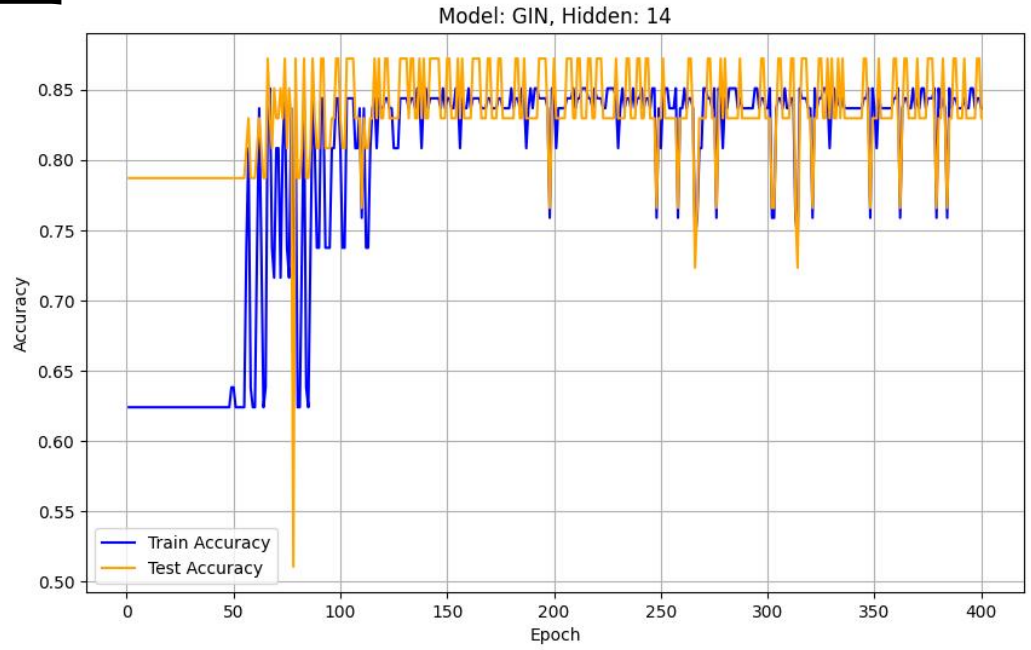

Model: GCN, Hidden: 64
Train Accuracy
Test Accuracy
Accuracy
Epoch


Model: GIN, Hidden: 16
Train Accuracy
Test Accuracy
Accuracy
Epoch


Model: GIN, Hidden: 32
Train Accuracy
Test Accuracy
Accuracy
Epoch

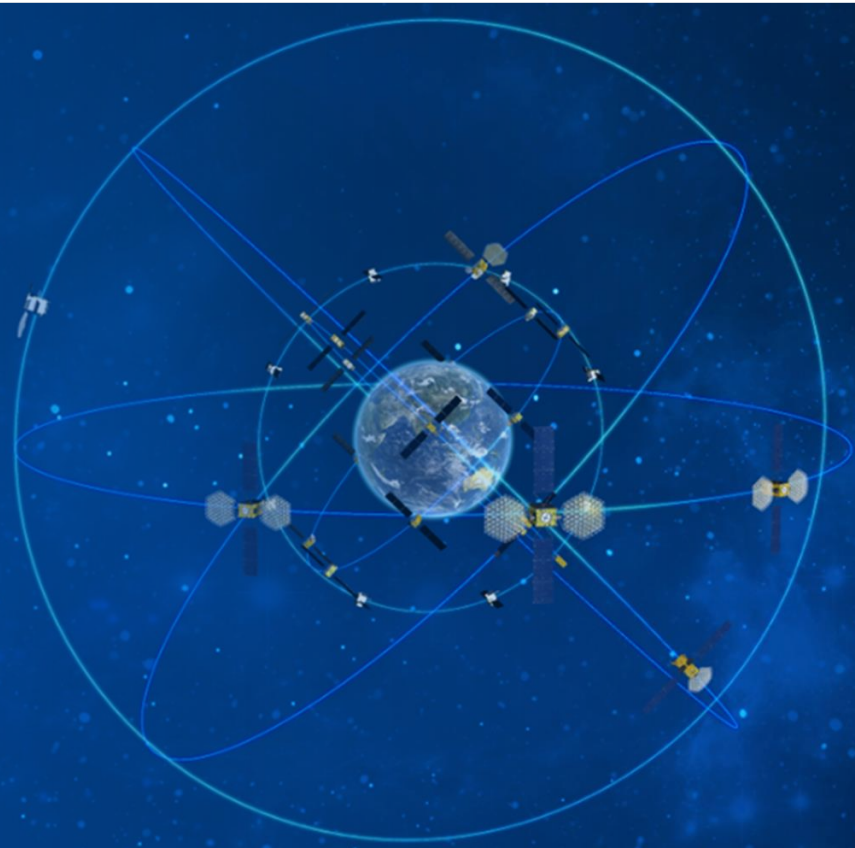
GIN模型


Model: GIN, Hidden: 64
Train Accuracy
Test Accuracy
Accuracy
Epoch


Model: GIN, Hidden: 14
Train Accuracy
Test Accuracy
Accuracy
Epoch

PART SIX

参考文献



[1] Siameh, T., "Semi-Supervised Classification With Graph Convolutional Networks," 2023. [Online]. Available: https://doi.org/10.13140/RG.2.2.22993.71526.

[2] K. M. Borgwardt and H. P. Kriegel, "Shortest-path kernels on graphs," *Fifth IEEE International Conference on Data Mining (ICDM'05)*, Houston, TX, USA, 2005, pp. 8, doi: 10.1109/ICDM.2005.132.

[3] M. Togninalli, E. Ghisu, F. Llinares-López, B. Rieck, and K. Borgwardt, "Wasserstein weisfeiler-lehman graph kernels," 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 2019, pp. 6439-6449, doi: 10.1109/CVPR.2019.00663.

[4] K. Xu, W. Hu, J. Leskovec, and S. Jegelka, "How Powerful are Graph Neural Networks?," *ArXiv*, abs/1810.00826, 2018.

代码仓库：https://github.com/seadeer-l/GNNmodels.git